JigsawHSI: a network for Hyperspectral Image classification

Jaime Moraga, H. Sebnem Duzgun

Abstract—**This article describes the performance of JigsawHSI,a convolutional neural network (CNN) based on Inception [1] but tailored for geoscientific analyses, on classification with the Indian Pines, Pavia University and Salinas hyperspectral image data sets. The network is compared against HybridSN [2], a spectral-spatial 3D-CNN followed by 2D-CNN that achieves state-of-the-art results in the datasets. This short article proves that JigsawHSI is able to meet or exceed HybridSN's performance in all three cases. Additionally, the code and toolkit are made available.**

Index Terms—**Hyperspectral Image Classification, Convolutional Neural Network, Remote Sensing, JigsawHSI, Pavia University, Indian Pines, Salinas**

I. Introduction

Hyperspectral image (HSI) classification is a classical task for remote sensing and machine learning practitioners, it consists in classifying the pixels from a hyperspectral image (HSI) into classes based on a given ground truth. For this tasks, several open source data sets have been released, including Salinas Valley, Pavia University [3], and Indian Pines [4].

Image classification, or semantic segmentation, is a machine learning task that has been tackled extensively in the literature, its application to HSI is an interesting problem because it is a difficult task for machines that humans can do efficiently. This allows for manual labeling of the images to create a ground truth, which makes the availability of data sets for supervised learning possible.

For machines, the task is complex because of the high dimensionality of HSI, and the spatial and spectral characteristics of the classification problem. This makes naïve approaches to the problem to be subpar, not achieving good results.

To solve this problem, many approaches exist in the literature, with hundreds of publications in 2022 alone. The first problem is high dimensionality, so different pre-processing algorithms have been proposed, for example dimensionality reduction by using decomposition functions like Principal Component Analysis (PCA), Factor Analysis (FA), Single Value Decomposition (SVD), and others [5]. Pre-processing steps also include the application of wavelet functions, Fourier transforms [6] and others.

(*Corresponding author: Jaime Moraga*, jmoraga@mines.edu)

J. Moraga is with the Department of Mining Engineering, Colorado School of Mines, 1610 Illinois St., Golden, CO 80401, USA

H. Sebnem Duzgun is the Fred Banfield Distinguished Endowed Chair and Professor, Mining Engineering, Colorado School of Mines, Golden, CO 80401, USA

Document created June 2, 2022

The task of image classification and semantic segmentation has been studied in machine learning and image analysis for decades. There are two main ways to approach the problem, by using pixel-based methods or area-based methods. In pixel-based methods, each pixel is classified independently of the surrounding pixels, this has the drawback of missing all the spatial and only analyzing the spectral information. A competing approach has been the use of both spatial and spatial information by analyzing cubes of data as a whole.

Since the seventies, neural networks have shown that they are specially well suited for this type of problem. Since the neocognitron [7], various shallow and deep neural and convolutional neural networks (CNN) have been used, with great success, to classify images with different number of bands (or channels) of data. In general, these have been limited to the red, green and blue (RGB) bands of visible light, but there is no reason to limit the analysis to just those three bands.

Among the CNN approaches applied to multispectral and hyperspectral data are 2-dimensional CNN and, lately, 3-dimensional neural networks. There have also been attempts to use hybrid approaches, as in the case of the HybridSN [2] which uses 3-dimensional kernels to both reduce dimensionality of the input and capture spectral information, and 2-dimensional kernels to extract spatial information (e.g. image textures).

Another network proposed for general multispectral and multi-variate data is the Jigsaw network, first used to evaluate the environmental impact of an iron mine dam collapse in Brazil [8]. This network has also been used successfully in identifying geothermal potential of two sites in Nevada (Brady and Desert Peak) [9]. This network is capable of identifying patterns both across the channel and spatial dimensions of an image.

This short article examines the use of a variant of the Jigsaw network, the JigsawHSI network, that is highly configurable in its hyper-parameters and depth to deal with a variety of inputs. The questions to answer are whether the network can tackle HSI classification problems, whether the results are comparable to more complex hybrid or 3-dimensional approaches, what dimensionality reduction functions can be used to achieve competitive results and what hyper-parameters are relevant to achieve such results. The network, configuration routines and sample configurations are made available to the public.

II. The JigsawHSI network

The JigsawHSI network is an adaptation of two networks based on the inception module by Szegedy et. al [1], depicted in Figure 1 . The first network was used to determine the

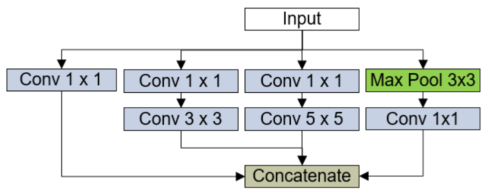

Fig. 1. Inception architecture

surface impact of a tailings dam collapse in Brumadinho, Brazil by classifying multispectral pixels from Sentinel-2 in before and after pictures, identifying the areas affected by the dam's iron ore waste tailings [8]. The second application was in the Geothermal AI where, by using multi-variate input layers (Mineral Markers, Temperature and Faults) that were the result of machine learning preprocessing, the network classifies pixels in an image as having or not geothermal potential [9]. In both cases, the inception network [1] was adapted to better capture the correlation in surface anomalies by increasing the number of internal filters of the inception network and adding a first 1x1 convolution after the input layer to better manage relationships across bands. The JigsawHSI's architecture builds upon this idea by both varying the number of internal $(n \times n)$ filters and making the first (1×1) convolutional layer optional, as shown in Figure 2.

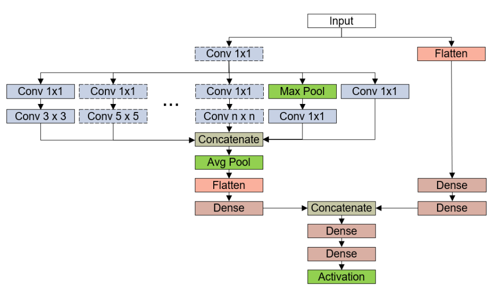

Fig. 2. JigsawHSI architecture. The dotted lined layers are optional

The network uses two parallel blocks that are merged in the second concatenate layer. The left side block uses the Inception-like module to capture spatial and spectral information finishing with a concatenate and average pooling layers, while the right side uses a flattening layer to discard spatial information and analyze all the spectral information of each pixel looking for linear and non-linear relationships. The bottom of the network has two fully connected dense layers and uses Softmax as the activation function, turning the results into classes.

This network's code and Jupyter Notebook are available in https://github.com/jmoraga-mines/JigsawHSI. The network is highly flexible, and can be configured by editing the `config.ini` file in the root directory.

The configuration includes the ability to define an analysis window that will partition the HSI $(width, height, bands)$ in squares of side k, creating cubes for analysis by the network of dimensions $(k, k, bands)$, a decomposition function can also be specified to reduce the dimensionality to c channels. The final input to the network will be of dimensions (k, k, c). Optionally, a layer with (1×1) convolutions can be added to perform filtering in the dimension of the channels (as in [10]), this starts the analysis in the spectral dimension of the HSI by the neural network, in this cases, the new value of c will be the number of filters defined for this step.

For training, several hyperparameters can be defined, these include the optimizer, learning rate of the optimizer, batch size, maximum number of epochs and a patience parameter to determine the number of epochs with no improvement required for early stopping of the training.

III. Configuration and training

The configuration parameters for training are in Table I for each of the datasets: Indian Pines (IP), Pavia University (PU), and the Salinas scene (SA) .

TABLE I
Experimental parameters for each JigsawHSI result

Parameters	Values IP	PU	SA
Window size	27	25	25
Decomposition	FA	SVD	FA
Input channels	9	9	12
Network design:			
HSI Filters	None	512	None
Filter size	9	9	7
Hyperparameters:			
Optimizer	Adadelta	Adadelta	SGD
Learning rate	0.1	0.1	0.01
Batch size	106	120	132
Max. Epochs	500	500	500
Max. Patience	20	40	20

Depending on the dataset, windows of different sizes can be optimal, in this case, IP uses a (27×27) window, while PU and SA use (25×25).

To simplify the analysis of the network, 4 decomposition functions can be used for dimensionality reduction: Principal Component Analysis (PCA), Functional Analysis (FA), Truncated Single Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), the number of final components can also be defined. For IP, FA and 9 factors are selected; PU uses SVD and 9 values; while SA uses FA and 12 factors.

An HSI layer with 512 filters is specified for PU. Therefore, the input's shapes will be $(27, 27, 9)$ for IP, $(25, 25, 512)$ for PU, and $(25, 25, 9)$ for SA.

The internal filters in the Inception-like network will go up to (9×9) for IP and PU, and (7×7) for SA.

The optimizers are also flexible, accepting Stochastic Gradient Descent (SGD), Adam and Adadelta; the learning rate of these optimizers is also configurable. The configuration used is $Adadelta(0.1)$ for IP and PU, and $SGD(0.01)$ for SA.

Batch sizes are 106 for IP, 120 for PU and 132 for SA. Maximum number of epochs is 500 for all cases. Patience is 20 for IP and SA, while PU requires to wait longer, with a patience of 40.

TABLE II
EXPERIMENTAL CLASSIFICATION METRICS (IN PERCENTAGES) FOR THE THREE DATASETS. OA: OVERALL ACCURACY, KAPPA: KAPPA FACTOR, AA: AVERAGE ACCURACY

Model	Indian Pines			Pavia University			Salinas		
	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA
HybridSN	**99.75 ±0.1**	**99.71 ±0.1**	**99.63 ±0.2**	99.98	99.98	99.97	**100**	**100**	**100**
JigsawHSI	99.74	99.70	98.11	**100**	**100**	**100**	**100**	**100**	**100**

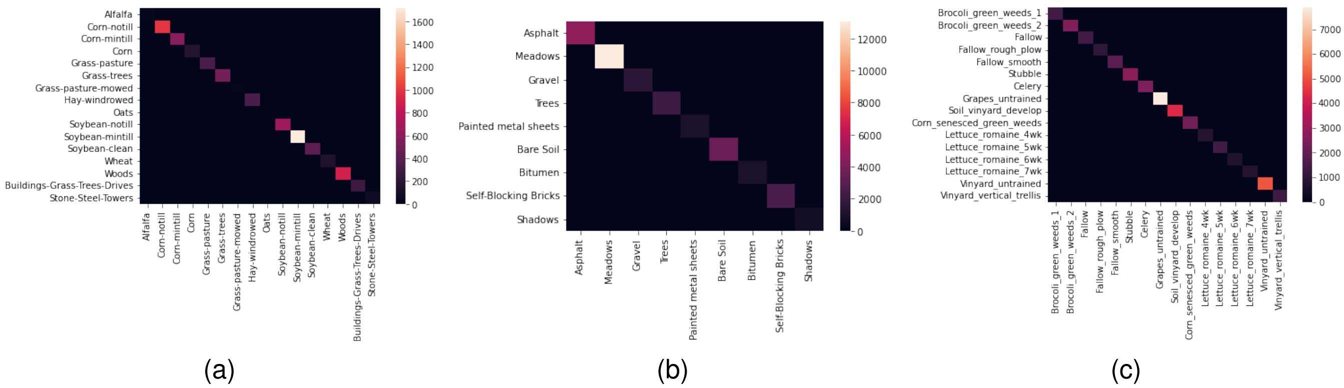

Fig. 3. Confusion matrix heatmaps for (a) Indian Pines, (b) Pavia U., (c) Salinas

The training was performed in a desktop computer running Ubuntu 20.4, with a single NVDIA card, Pyhton 3.8 and the latest version of CUDA.

For comparison, we used HibridSN (https://github.com/gokriznastic/HybridSN, published in IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 17, NO. 2, FEBRUARY 2020). Both networks were trained using 30% of the samples for training and 70% of the samples for testing.

IV. EVALUATION AND DISCUSSION

The convolutional neural network (CNN) is one of the most used and successful networks in computer vision and visual data processing. Traditionally, these networks have been designed with 2-dimensional (2D) CNN filters due to the datasets having relatively shallow three band (RGB) images, and in this avoiding the curse of dimensionality. In the case of HSI, even after dimensionality reduction, the number of bands or channels are significantly larger than three.

HybridSN is a hybrid spectral-spatial 3-dimensional (3D) CNN followed by a a spatial 2D-CNN [2]. In theory, this should ease the representation of joint spatial-spectral information by the network. The HybridSN network was tested on all three datasets and results compared to five additional networks, achieving state-of-the-art results.

In Table II, we can see that the JigsawHSI obtains results that are equivalent or better than HybridSN. In the case of Indian Pines, JigsawHSI achieves lower scores in overall accuracy (OA), Cohen Kappa (Kappa) and average accuracy (AA), but in the first two cases the difference is near 0.01% and well within the margin of error of HybridSN. For the Pavia University dataset, the JigsawHSI obtains better results than HybridSN, achieving a 100% accuracy for the test set. In the case of Salinas, both networks obtain the maximum accuracy.

In Figure 3 the confusion matrix heatmaps show that in all cases the accuracy was perfect or almost perfect.

This is a surprising result at first sight because the use of 3D-CNN for spectral analysis cannot be replicated in full by the JigsawHSI. This implies that either the 3D-CNN is not needed for this case, or the 3D-CNN is not helping the spectral representation of the image. We theorize that by using dimensionality reduction as a first step in both networks, the 3D-CNN is not needed to represent in full the relationships between channels.

The other anomaly is in the AA score. This comes as a result of JigsawHSI not being able to discriminate well the oats samples in IP. This is most probably caused by the sparsity of the class, where only 14 test samples are provided.

In general, the JigsawHSI is able to match or improve on the results of the HybridSN.

V. CONCLUSION

In this article, the JigsawHSI network was introduced, a modification of the Jigsaw network and Geothermal AI networks, both based on the Inception architecture. The network was tested against the Indian Pines, Pavia University and Salinas Valley datasets, three freely datasets widely used in hyperspectral image classification.

The JigsawHSI network achieves results that are either equivalent or better than HybridSN under the same conditions, showing that with 2D-CNNs results match a more complex 3D-2D-CNN when dimensionality reduction is applied as a first step.

Additional research directions include to test of the networks using smaller test samples, eliminating the dimensionality reduction and using spatial cross validation to better assess the generalization capabilities of both networks.

ACKNOWLEDGMENTS

Hyperspectral datasets collected by M Graña, MA Veganzons, B Ayerdi and provided by Grupo de Inteligencia Computacional (GIC) (https://ehu.eus/ccwintco). Pavia scenes originally courtesy of Prof. Paolo Gamba and Prof. Fabio Dell'Acqua from the Telecommunications and Remote Sensing Laboratory, Pavia university (Italy). Special thanks to Gopal Krishna, for making HybridSN available in Github.

REFERENCES

[1] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11231

[2] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 277–281, Feb. 2020, conference Name: IEEE Geoscience and Remote Sensing Letters.

[3] P. Gamba, "A collection of data for urban area characterization," in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 1, 2004, p. 72, pavia Universitt dataset.

[4] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3," Sep. 2015, doi:10.4231/R7RX991C. [Online]. Available: https://purr.purdue.edu/publications/1947/1

[5] Z. Meng, L. Jiao, M. Liang, and F. Zhao, "A Lightweight Spectral-Spatial Convolution Module for Hyperspectral Image Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[6] A. V. Miclea, R. Terebes, and S. Meza, "Local binary patterns and Fourier transform based hyperspectral image classification," in *2020 International Symposium on Electronics and Telecommunications (ISETC)*, 2020, pp. 1–4.

[7] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. [Online]. Available: http://link.springer.com/10.1007/BF00344251

[8] J. Moraga, G. Gurkan, and P. D. H. S. Duzgun, "Monitoring The Impacts of a Tailings Dam Failure Using Satellite Images," in *USSD Elevate Conference 2020*, vol. 2020. Denver, CO: United States Society on Dams (USSD), Apr. 2020. [Online]. Available: http://ussd.conferencespot.org/2020/pdf/a042/0360_0531_000071

[9] J. Moraga, H. S. Duzgun, M. Cavur, and H. Soydan, "The Geothermal Artificial Intelligence for geothermal exploration," *Renewable Energy*, vol. 192, pp. 134–149, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S096014812200581X

[10] M. Lin, Q. Chen, and S. Yan, "Network In Network," arXiv, Tech. Rep. arXiv:1312.4400, Mar. 2014, arXiv:1312.4400 [cs] type: article. [Online]. Available: http://arxiv.org/abs/1312.4400

VI. BIOGRAPHY SECTION

Jaime Moraga is a PhD student at Colorado School of Mines, Golden, CO

H. Sebnem Duzgun H. Sebnem Duzgun is the Fred Banfield Distinguished Endowed Chair and Professor, Mining Engineering, Colorado School of Mines, Golden, CO

VII. Appendixes

Appendix
Indian Pines results

```
Accuracy by target (in percentages):
100.0000 : Alfalfa
 99.0000 : Corn-notill
100.0000 : Corn-mintill
100.0000 : Corn
100.0000 : Grass-pasture
100.0000 : Grass-trees
100.0000 : Grass-pasture-mowed
100.0000 : Hay-windrowed
 71.4286 : Oats
100.0000 : Soybean-notill
 99.8255 : Soybean-mintill
 99.5181 : Soybean-clean
100.0000 : Wheat
100.0000 : Woods
100.0000 : Buildings-Grass-Trees-Drives
100.0000 : Stone-Steel-Towers
```

Appendix
Pavia University results

```
Accuracy by target (in percentages):
100.0000 : Asphalt
100.0000 : Meadows
100.0000 : Gravel
 99.9534 : Trees
100.0000 : Painted metal sheets
100.0000 : Bare Soil
100.0000 : Bitumen
100.0000 : Self-Blocking Bricks
100.0000 : Shadows
```

Appendix
Salinas Valley, CA results

```
Accuracy by target (in percentages):
100.0000 : Broccoli_green_weeds_1
100.0000 : Broccoli_green_weeds_2
100.0000 : Fallow
100.0000 : Fallow_rough_plow
 99.9467 : Fallow_smooth
100.0000 : Stubble
100.0000 : Celery
100.0000 : Grapes_untrained
100.0000 : Soil_vineyard_develop
100.0000 : Corn_senesced_green_weeds
100.0000 : Lettuce_romaine_4wk
100.0000 : Lettuce_romaine_5wk
100.0000 : Lettuce_romaine_6wk
100.0000 : Lettuce_romaine_7wk
100.0000 : Vineyard_untrained
100.0000 : Vineyard_vertical_trellis
```